Graham Thompson

'Frank Lloyd Oop': Microserfs, *Modern Migration, and the Architecture of the Nineties*

'Frank Lloyd Oop': Microserfs*, Modern Migration, and the Architecture of the Nineties*

I

If the early development of the computing industry in America was marked by a preoccupation with hardware, as companies like UNIVAC, DEC, and IBM filled the nation's corporate and government offices with mainframes, then a similar preoccupation has so far marked the response of cultural criticism to contemporary technology. For Michael Menser and Stanley Aronowitz, American technoculture is founded on the way that hardware permeates all sections of society: 'The Amish have their wagons and farm equipment, the hippies their Volkswagen buses. The rap DJ has his or her turntable … the cyberpunk has a computer complete with modem' (Menser and Aronowitz, 1996: 10). Even in a recent article about the interaction between people and computers, Kevin J. Porter treats the computer, without exception, as a piece of machinery (Porter, 2000). Software—the medium through which human-computer interaction takes place—is nowhere to be found in either of these accounts.

According to Paul Ceruzzi, however, a watershed has been reached in the relative economic significance of hardware and software. During the 1970s and 1980s, as personal computers started appearing on people's desktops at work and at home, the software market began to be fully exploited by companies like Microsoft. By the 1990s the development and marketing of software—a category of product, Ceruzzi reminds us, that 'by definition, has no essence'—had started to overshadow the hardware that 'was becoming in some cases a cheap mass-produced commodity' (Ceruzzi, 2000: 79).

The cultural impact of living in a software age like this is precisely what concerns Douglas Coupland in *Microserfs*. And it is Abe, work colleague and housemate of the novel's narrator Daniel Underwood, who pinpoints the particular spatial consequences of this change. One day Abe complains to Dan about the architecture of the 1990s: 'He said that because everyone's so

poor these days, the '90s will be a decade with no architectural legacy or style—everyone's too poor to put up new buildings. He said that code is the architecture of the '90s' (Coupland, 1995: 23. All further reference appear in parentheses). By code Abe means computer code. This is what the occupants of the group house in which he lives deal with all day in their jobs at Microsoft, the writing of it and the testing of it; the software that is produced from it has made Bill Gates the richest man in the world. And although Abe may not be poor—he is the 'in-house multimillionaire' (5) thanks to his work as a coder—he has no desire to own, let alone build, his own property. He rents his room instead, just like the others, and in this regard is well-placed to draw attention to one striking fact: for people working in the software industry, and for an increasing number of the rest of us too who live in a software age, the most important forms of architecture are migrating from the spaces that surround us—homes, workplaces, public buildings—to the code that is invisible and mostly incomprehensible to us inside our computers.

I want to focus on this kind of migration here not only to redress the critical neglect of software but because of the opportunity it offers to refine another narrative of migration that often gets told about the development of the computing industry in the last twenty five years. It is possible to guess the spirit of this other narrative when Bill Gates, in *The Road Ahead*, talks about computing as a journey that 'has led us to places we barely imagined' and that one of the major forces for economic progress in the next millennium will be the 'Internet Gold Rush' (Gates, 1996: xii, 262).[1] Not even the domesticated disguise of his casual chinos, and open-neck shirt and sweater on the front cover of the book would seem to mask the fact that the words of the generation's wealthiest businessman have evolved directly from a discourse about American history that is bound up with the frontier: its establishment, its breeching, and its displacement into the realms of technology. The metaphors of the road, the journey, and the Gold Rush are not myths then; they are the conditions that compelled Gates and Paul Allen to migrate to New Mexico in the 1970s, the conditions that compelled the migration of thousands of people to California in the 1840s, and the conditions that compelled a similar migration to California in the

1990s of people who wanted to get their computer code onto the most important road in the world, the information superhighway.[2]

It is just such a journey that Dan and his housemates make once they decide to leave Microsoft and head for Silicon Valley. They are joining Arthur Kroker's 'Virtual Class', a new economic grouping, he argues, that is 'compulsively fixated on digital technology as a source of salvation from the reality of a lonely culture' (Kroker, 1996: 168). Authoritarian, anti-democratic, against political dissent, and against aesthetic creativity, Kroker suggests that the information superhighway 'represents the disappearance of capitalism into colonized virtual space', this virtual colonialism representing, he suggests in apocalyptic terms, 'the endgame of postcapitalism' (Kroker, 1996: 170, 178).

The problem with Kroker's assessment of shifts in the dimensions of capitalist activity, is that as a critical position it can never account for the way in which these shifts are experienced, either by the people directing them or upon whom they are exacted. People like Abe for instance, who, despite the millions of dollars he has made from computing by the age of thirty, has 'nothing to his name but a variety of neat-o consumer electronics and boxes of Costco products purchased in rash moments of Costco-scale madness' (10-11). This kind of lifestyle is one that remains outside the scope of Kroker's attention. More helpfully, Peter Stoneley has noted that while the most keenly-sought transformation of gold-rush hopefuls in the nineteenth century was the change from being poor to being rich, the literature of those who experienced such attempts often points 'toward a much more general sense of change and disorientation' (Stoneley, 1996: 189). It is just such a sense of change and disorientation in *Microserfs* that interests me. It is a novel that takes on and intersects with the narrative of pioneering entrepreneurism only to render that narrative unfocused and imprecise. Dan, in fact, touches on the transitional nature of this experience when he is persuaded to keep his diary more regularly by his girlfriend: 'Karla got me to thinking that we really *do* inhabit an odd little nook of time and space here, and that odd or strange as this little nook may be, it's where *I* live—it's where *I am*' (63).

What interests me here, then, is a migration that has seen the products responsible for Gates's huge wealth—computer software built from millions of lines of code—begin to alter the very nature of traditional capitalist living and working environments to such an extent that the most important office in people's working lives is no longer the office building in which they work and leave to return home, but the copy of Microsoft Office in all its binary glory that they have loaded on their desktop computer ... and on their portable laptop … and on their home computer and that can follow them from place to place. It is the consequences of change like this, I would argue, that contribute toward 'the general sense of change and disorientation' that is experienced by the characters in Coupland's novel as their lives cross the path of this particular historical moment.

II

In June 1996 the *San Jose Mercury News* ran a feature entitled 'Sleepless in Silicon Valley' about the working habits of people employed at local computing companies (Leibovich, 1996). Printed with the article was a series of photographs taken by Meri Simon. One showed a computer programmer covered by a blanket asleep under his desk in a cluttered office. Papers are strewn across the floor and work surfaces, and a soccer ball lies at the sleeper's feet. This latter-day Bartleby is not in danger, however, of angering his employer. He *is* the employer. The photograph is of David Filo.[3] At the time the picture was taken he was worth five hundred million dollars. As the co-founder of Yahoo!, the internet directory service, he is now worth several times that amount. Yet he no longer has his own office. When Po Bronson visits him he finds him sharing a double cubicle with one other person, and 'a trash heap of paper … forty inches deep of unread memos, promotional literature, office chatter. … It was his inbox and filing system'. For Bronson the irony of this trash-heap is that 'the guy who has engineered the

most popular directory for organizing the morass of the World Wide Web' is someone who is 'utterly unable to engineer an organizational system for his own paper flow' (Bronson, 1999: xv).

In this working world, then, offices have clearly changed. No longer are they the places where one spends a specific portion of the day and from which one then routinely retreats to the private world of the home. Offices are where one may live for days at a time. Part of the impetus for the alteration in this usage pattern derives from the difficulty in allocating a place for the writing of code and the development of software applications. Is it work, or is it a hobby and a leisure activity that, for these code-writers, pre-dates its being turned into work?[4]

It is this kind of working world that Dan and his colleagues occupy at Microsoft, a code factory where the ability to 'narrow-focus' makes nerds 'so good at code-writing: one line at a time, one line in a strand of millions' (2), but where the primacy of this ability means that traditional requirements of office organization can be ignored: 'the campus is utterly casual' (25) according to Dan. There are no restrictions on the hours that he keeps or how he organizes or decorates his office. The office corridors are lined with '*Far Side* cartoons taped to windows, Pepsi can sculptures taped to the walls, and inflatable sharks hanging from the ceilings', and life just would not be the same without the 'weekly-ish communal stress-relieving frenzies' that on one occasion consists of punishing 'plastic troll dolls with 5-irons, blasting them down the hallway, putting yet more divots in the particle board walls and the ceiling panels' (30).

This is behaviour that is not just tolerated by the Microsoft management structure—a structure that by the standards of the 1960s and 1970s corporation is streamlined to a skeletal degree—but an attitude to work that is fostered by the company as early as the hiring stages of employment. Advice about self-presentation for job interviews seems to make no sense in the computing software world. Fred Moody describes one interview candidate at Microsoft: 'he was wearing a baggy striped T-shirt, boat shoes without socks, and oversized shorts made by tearing the legs off a pair of sweatpants' (Moody, 1995/1996: 124-5). The interviewer, with rock radio playing in the background, conducts the interview seemingly oblivious to dress and purely on the

candidate's capacity to solve coding problems on a whiteboard. It is code that counts, the ability to narrow focus, to block out the surrounding architecture and immerse oneself in the architecture of the code.[5] So important to a company like Microsoft are these work values that, just as with David Filo, what matters now is not an ability to organize and control the space in which one works, but the code that will organize the information with which end-users will interact and perform their work or leisure.

Coupland dramatizes this change by having IBM exist in his novel as the embodiment of all that Microsoft has superseded. 'It must be so weird', Dan notes, 'living the way my dad did—thinking your company was going to take care of you forever' (17). Dan's father has worked for IBM since he left his job in education in the mid-1980s, although he is sacked not long into the novel. Susan, another of the housemates, is an 'IBM brat and hates the company with a passion. She credits it with ruining her youth by transferring her family eight times before she graduated from high school … nothing too evil can happen to IBM in her eyes' (9). Camped up in Redmond, Microsoft makes none of these disruptive demands on its employees. It makes them work long days—'In at 9:30 A.M.; out at 11:30 P.M.', or sometimes 1:30 A.M. or 2:30 A.M.—but instead of making them move home, Microsoft allows its employees to move their homes inside their own offices, and even provides 'employee kitchen[s]' full of 'dairy cases of Bill-supplied free beverages' (16). The result is that working at Microsoft Dan finds that his 'weekends are no different than [his] weekdays' (18). One's experience of time alters as a consequence:

> Today, while raking the front lawn, Todd said, 'Wouldn't it be scary if our internal clocks weren't set to the rhythm of waves and sunrise—or even the industrial toot—but to *product cycles*, instead?'
>
> We got nostalgic about the old days, back when September meant the unveiling of new car models and TV shows. Now, carmakers and TV people put them out whenever. Not the same. (55)

Todd has identified what has happened, even though it may be too scary for him to admit. This erasing of the cycles and the time clock of industrial capitalism, along with the erasing of the notion of career trajectory—either within one firm or between firms (which company would want a coder after Microsoft has had their best years?)—marks an important change in the way that capitalism interacts with its labour. Microsoft encouraged this change with its use of stock options when it was employing people in the 1980s. While one might see it as a way of exacting loyalty from staff, the reality has been—and this is due to Microsoft's success—that once the stock has been held for the necessary amount of time, the employees then cash in, sell their stock and leave Microsoft. They celebrate with 'vesting parties', just like Susan does. She quits Microsoft the day after she 'vests' and unveils her new image to her housemates. Her previous image, 'Patagonia-wearing Northwest good girl—had been shed for a radicalized look: bent shades, striped Fortrel too-tight top, Angela Bowie hairdo, dirty suede vest, flares, and Adidases' (62).

This possibility of vesting into richness and identity change, however, is beginning to fade at Microsoft. As well as this, there are other factors that make Microsoft a less appealing place to work than it once was and that makes an escape to Silicon Valley more enticing. Coupland's novel is, after all, called *Micro*serfs. The relentless work into the early hours of the morning is one part of the drudgery, but the working environment is also as alienating and surveilling[6] as it is tolerant and unsupervised. It still manages to institute an almost *1984*-like sense that one is being watched. By Bill.

All routes in Microsoft lead back to Bill Gates. Gates is noted not just for his tantrums in meetings at Microsoft, but also for his stunning technical ability that means he has a different relationship with his employees and their work. As one employee says, 'He'll know some intricate low-level detail about a program, and you wonder, 'How did he know that? He has no reason ever to get to that level!' Some piece of code, or some other technology that Microsoft isn't even involved in. You just shake your head' (Moody, 1995/1996: 80). These exacting standards mean

that one must prepare as precisely as possible for encounters with him and this need for preparation is passed along the management line. Consequently Microsoft becomes a company dominated by the image of one man, 'Citizen Gates' as he has become known.

Indeed *Microserfs* begins with Michael, another of Dan's housemates, receiving flame-mail from Bill. And the subject of the mail is 'a chunk of code Michael had written', about which Bill just 'wailed on'. E-mails like this work to train one's mind on what one is doing, a classic tactic to instil disciplinary self-surveillance, and as Dan points out 'We figured it must have been a random quality check to keep the troops in line' (1). But what is telling about this e-mail is the spatial impact it has on Michael. He locks himself in his office and refuses to come out. Dan gets so concerned that in the middle of the night he drives to the Safeway store to buy flat foods to push under Michael's door—'Kraft singles, Premium Plus crackers, Pop-Tarts, grape leather, and Greezie-Pops' (2)—for which Michael is grateful when he finally emerges from his office the next day after sleeping in there overnight. He determines not to eat anything that is not two-dimensional thereafter: 'Ich bin ein Flatlander', he declares, in a jokey but compelling way that at first seems to position him and his Microsoft cohorts in a trajectory of American national technology that was heightened by a President who not only claimed that 'Ich bin ein Berliner' but also sanctioned the 'space' race. Once more, though, the two narratives of migration intersect contrapuntally, since it is not the frontier of space and other-earthly exploration that concerns this generation, but a textual, coded world that exists—like flat food—in two dimensions, even though on a computer screen it can come to mimic a three-dimensional environment. This mimicking is the project that Michael—and it is significant that it is Michael—ultimately embarks upon and on which the rest of the housemates join him.

Working at Microsoft, then, gives something of a hint of the nature of how the office environment has changed in the new computing software industry. And yet the company's success, which has led to its increasing size,[7] has also led to the negation of the innovative strategies that threatened to turn Microsoft into the 'home' of a new working ethic. The road

ahead for Dan and his colleagues and friends does not involve Microsoft. There is no doubt that Microsoft and Silicon Valley in *Microserfs* come to represent different working environments. It is not the pursuit of code that Dan objects to *per se*, since his new work in Silicon Valley will be the pursuit of more code. It is code that represents 'somebody else's abstraction' (90-91) that he resents; an abstraction that becomes solidified in the monolithic Microsoft. The code that is written there is enshrined in the Campus and in the Microsoft branding that connects everything to Bill. Silicon Valley, on the other hand, is a place that exists as a visual phenomenon on the same level as the computer code on which its importance as a place is built: it is invisible. But, as Dan himself says, 'invisibility is invariably where one locates the ACTION' (137).

III

The failure of traditional visual techniques to sufficiently capture the nature of Silicon Valley is one way of beginning to think about the status of this invisibility and about the migration of a physical architecture to an architecture based on code. Following a film crew trying to make a documentary about the place and about the computing industry, Po Bronson finds that after a whole day's shooting they still haven't found an establishing shot, something that would serve a similar purpose to the letters on the Hollywood hillside telling the audience where they are. Part of this failure, I think, derives from the incoherent intersection of the two migratory narratives I defined earlier. Slotted into a traditional frontier narrative, Silicon Valley is a story of high stakes, hard work, sudden wealth, and rapid growth, conditions that would seem to offer copious opportunities for synecdochic visual images, and yet in Silicon Valley all the film crew find is 'an endless suburb, hushed and nonchalant, in terrain too flat to deserve the term "valley"' (Bronson, 1999: xvii). There is no Valley architecture to represent to the world the vertiginous impact this location is having upon people's lives, nothing to mark it as a distinctive place with a distinctive relationship to a style of capitalism.

Yet if one follows that other narrative of migration I outlined it is possible to discern instead that here is an industry that marks itself in an altogether different kind of way and in an altogether different place: it is on one's desktop in the 'screenful of icons that make computers touch-feely familiar' (Bronson, 1999: xvii) that the Valley appears. The architecture of Silicon Valley, then, resides in the very code it produces. And by migrating its architecture in this way—from the three-dimensional to the two-dimensional, from the material to the hieroglyphic—Silicon Valley has facilitated the flattening of the distinctions between the workplace and the home, between work and leisure, that have stood at the heart of the experience of work in American culture in the last hundred years. Once the architecture of code replaces the architecture of the built environment as the site for the creation of value in a capitalist economy, the need for the workplace to be discretely marked and separated becomes less and less important. At Netscape a dentist visits the office site several times a week so the employees there don't have to leave work to take care of their teeth; at Excite they have office laundry facilities for workers who don't have the time to do their washing at home. The office park—combining a work environment with a mall environment, a producing with a consuming environment, a work with a leisure environment—has become a Silicon Valley phenomenon that, according to Bronson, is part of a whole design to blur the distinction between work and non-work, between indoors and outdoors, and between work and rest. Silicon Valley 'is this concept taken to the level of a whole region: it's one big office park' (Bronson, 1999: xviii-xix). At the same time, the migration of architecture in this way provides a means for exporting the structures of this architecture to every desktop across the computerized world.

It is into this invisible powerhouse of American capital that Dan and his Microsoft colleagues move. Crossing the border into California from Washington, Karla remarks that "'We live in an era of no historical precedents … The cards are being shuffled; new games are being invented. And we're actually *driving* to the actual card factory'" (99). The double use of 'actual' here—since they are not driving toward a card factory at all—draws attention to the metaphor Karla uses and

so to the underlying virtuality of Valley economics. For Dan, the world of Silicon Valley, while it may have no historical precedents, certainly exists in a continuum of capitalist development. Charting the shift in the relationship between corporation and employee since the 1970s he notes the gradual 'integration of the corporate realm into the private' (211) as corporations provided workplace sweeteners—showers for lunchtime joggers, sculptures—that attempted to 'soothe the working soul'. The campus model at Microsoft and Apple was the next stage in this process, when 'the borderline between work and life blurred to the point of unrecognizability' (211). The final stage of this process is Silicon Valley in the 1990s: 'corporations don't even hire people anymore. People become their own corporations' (211). This transition—or delegation—of the corporate ethos also marks a transition in the location of the production of history. Dan's father belongs to that generation who believed that 'history was created by think tanks, the DOE and the Rand Corporation of Santa Monica, California' (203-4); the same paranoid generation for whom big business and big government epitomised the control over the individual of the military-industrial complex. Whilst re-visiting the IBM plant after he has been fired, Dan's father says, 'I never thought history was something my kid built in the basement. It's a shock' (204).

Within this context, then, and as if to emphasize the collapsing distinction between the workplace and the home that the shift to a virtual architecture of code helps bring about, Dan's parents' Palo Alto house becomes the place where Dan, Karla, and Michael live and where the new business venture is located. The product they work on is a game developed by Michael called *Oop!*. This game, not coincidentally I would argue, is a 'virtual construction box'; a kind of computerized version of Lego. But whereas Lego bricks have only a small number of 'bumps' that can be connected to other Lego bricks, *Oop!* bricks can have thousands of bumps and so the possibilities for creating complex objects is vastly increased:

Imagine:

> **'Oopenstein'**—flesh-like ***Oop!*** bricks or cells, each with ascribed biological functions that allow users to create complex life forms using combinations of single and cloned cell structures. Create life!
>
> **'Mount Oopmore'**—a function that allows users to take a scanned photo, texture map that photo, and convert it into a 3D visualized ***Oop!*** object.
>
> **'Oop-Mahal'**—famous buildings, preconstructed in ***Oop!***, that the user can then modify as desired.
>
> **'Frank Lloyd Oop'**—architectural ***Oop!*** for adults (71).

In computing terminology OOP is an acronym of Object Oriented Programming, a particular kind of approach to writing code for contemporary Windows software applications. One of the main benefits of the Windows operating system is that it is device-independent. It separates the specific hardware devices on a computer—the keyboard, monitor, hard disk drive, mouse—from the software programs that run on it. This separation is known as abstraction and it allows software programmers to work with general categories of hardware rather than specific makes and models. They can write code for whatever printer is attached to the computer system, or whatever monitor or keyboard. Windows—through model-specific drivers for particular hardware devices—does the job of letting the software application communicate with the computer hardware. Object-oriented programming is important because, as its name suggests, it breaks the programming process down into a series of objects that can be reused, not only in the same software application, but in other applications too. These reusable objects shorten development time, are easily distributable, and facilitate the kind of group coding projects that one finds not only at Microsoft but on Michael's *Oop!* project where each person is given a discrete task. In one way, then, the division of labour that object-oriented programming requires could be seen as a classic *laissez-faire* or even Taylorist solution to software production.

Oop!'s relation to object-oriented programming is left unexplained in *Microserfs* but I think that what the use of this acronym suggests is that *Oop!* serves the purpose in the novel of something more than just another computer game on the market. It has a symbolic value, I would suggest, that allows the binding together of the various thematic elements of the novel—work, technology, capitalism, computing—around this central motif of code as the architecture of the 90s. As the very reason why Michael and Dan and their colleagues migrate to Silicon Valley, it stands as a powerful literary metaphor foregrounding the idea that the ultimate purpose of code is to produce architecture, although not the kind of architecture that will occupy the traditional spaces of the body (Oopenstein/Frankenstein), the national monument (Mount Oopmore/Mount Rushmore), the tourist attraction (Oop Mahal/Taj Mahal), or the workplace (Frank Lloyd Wright's most famous office design was for The Larkin Building in Buffalo, New York, 1904). Instead it is an architecture that has migrated into the syntax of code and thus it is here that one should look if one wants to know about work, technology, capitalism, and computing in the 1990s.

Code, then, is the synecdochic image the film crew are looking for. But code is flat and two-dimensional; it is text and not images. It is the failure to recognize that the dazzling visual effects of a program like *Oop!,* of a place like Silicon Valley, are produced not by pioneering technological frontiersmen and their visual baggage but by strictly regulated strings of commands, functions, and syntax in coded textual information, that leaves the film crew stumbling around for that one image which they hope can capture the essence of the Valley. In a real sense this image actually resides within the computerized equipment they carry with them during their search.

It is also worth stressing here just how important the link is between computer code and an architectural terminology that draws on familiar aspects of office design. The key moment in this development, and the key moment that enabled control over computing software to be passed on to the employee and the consumer, was the shift from a character-based to an icon based

command structure. This development was pioneered by Apple, but is dominated now by Microsoft. This was the shift from MS-DOS to Windows operating systems. Before the arrival of the Windows platform, IBM-compatible PC users were forced to type in 'often-obscure commands', as Gates himself puts it, in order to run programs and get these programs to do anything. The thinking behind Microsoft's development of Windows was to create an easier to use interface in order 'to realize our vision of widespread personal computer use' (Gates 57).

The Windows of any contemporary office, rather than offering views of the world outside, will now offer ever-changing views of all kinds of spreadsheets, databases, word-processed documents, files, folders, and messages. What this new Windows environment facilitates—the new Windows environment that can contain the operations of a whole office—is the migration of huge amounts of information from disparate physical locations to the few square inches of real estate that is a computer monitor, and which is accessed by gazing through a myriad of Windows, since every part of every program interface—the scroll bars, the toolbars, the icons, the menus—is a separate window seamlessly integrated into one big window that contains the program. Concentrated and compacted, this information sits waiting as a collection of binary code until it is given a form and an intelligibility by Windows.

The importance of Silicon Valley for me in *Microserfs* is how its relative invisibility can be seen to be both a consequence of, and a metaphor of, the changing site of economic productivity and value creation in contemporary capitalism; the shift from a visible architecture of material production to an invisible architecture of code. The principle on which software products and Silicon Valley are founded—code as architecture—has literally altered the epistemological organization of capitalist working space.

V

It is an engagement with code and encoding that beats away at the heart of Dan's narrative in *Microserfs*. The first hint we get that Coupland is drawing attention to it is early on when Todd invents a program called 'Prince Emulator' that converts 'whatever you write into a title of a song by Minnesotan Funkmeister, Prince' (18). Dan rewrites part of his diary with it:

> A few minutz l8r I bumpd in2 Karla walkng akros the west lawn. She walkz rely kwikly & she'z so small, like a litl kid (18).

'I reread the Prince Version', he says, 'and realized th@ after a certain point, real language decomposes into encrytpion code; Japanese' (19). Dan then begins to keep a file on his computer called 'Subconscious' that consists of random words that come into his head, and which then begin to appear at the end of his diary entries, creating almost a second level of narrative which might be read as a condensed—although alternative—account of that day or week's diary entry (see 46, 49, 52 for examples). It isn't just Dan, either, who helps to define this relationship between code, the software computing industry of the 90s, and the people experiencing this world. Ethan, the businessman Michael teams up with to help raise venture capital for the *Oop!* project, first meets Michael 'inking out all of the vowels on his menu' in a diner; Michael explains that he was "'*Testing the legibility of the text in the absence of information*'" (109). Dan copies this attempt and converts another of his diary entries (308); and then converts it so it has no consonants.

These entries allow a glimpse at the workings of Dan's techie, 90s-acculturated mind, and it is clear that this is sometimes not a contented place. The word 'Windows' translates into 'Prozac' within the space of a few lines (182). And Coupland even takes to giving the reader encoded messages in his text that hint at the potential dangers of computing technology. On pages 104 and 105, what appears to be a random series of 0s and 1s actually turns out to read:

> I heart Lisa Computers
>
> This is my computer

There are many like it,

but this one is mine.

My computer is my best friend.

It is my life.

I must master it,

as I must master my lif%.

Without me, my computer is useless.

Without my computer, I am useless.

I must use my computer true.

I must com}ute faster than my enemy who is trying to kill me.

I must outcompute him before he outcomputes me.

I will.

Before God, I swear this creed.

My computer and myself are defenders of this country.

We are the masters of our enemy.

We are the saviours of my life.

So be it until there is no ene-y, but peace.

Amen.

Tinned peaches

Yttrium

San Fran[8]

This coded message seems to suggest that harnessed in the interests of the state, the contact of person and computer only enhances such interests. The cult of Bill outlined by the novel also clearly suggests the way that computing technology is implicated in the consolidation of economic power and capital. And yet it would be hard to draw such causative associations from these messages, especially since *Microserfs*, I would suggest, is a novel that demands attention not as an arbitrator between two competing schools of thought about computing technology—that

is, technology is good/technology is bad—but as a novel that represents the territory that is left unexamined by such a polarized debate: the way in which technology is experienced.

My point here is that in this strange transitional landscape that Dan and his friends are negotiating, code has become not only the engine driving capitalist accumulation and economic expansion, but the very means by which communication proceeds and the way, therefore, by which we—following Dan—might understand how meanings and institutions might sometimes be at odds with one another. Code is important in Coupland's narrative because, by being in such close relation both to the industry and the economy that relies upon it *and* to the non-visual form of communication that people are increasingly using, it stands as the mediating object through which the people in that economy try to come to terms with the 'general sense of change and disorientation' that accompanies the experience of important economic change. This disorientation may manifest itself as 'little fears' as it does for Dan: 'fear of not producing enough; fear of not finding a little white-with-red-printing stock option envelope in the pigeonhole' (38). It may manifest itself in the 'Perfectville' train set landscape that Dan's father builds; or in the gay Bug's desperation to 'find a niche' (306); or in Dan's anxiety about the diminishing distance between man and machine (228). Code, because of its encrypted nature, signals in an ambiguous way and this is why *Microserfs* is a novel, I want to suggest, where things do not line up. Work and leisure have stopped being discrete domains of one's life; the workplace and the home can now be the same place, or different places that look the same; architecture is now not only visible but invisible as well, a virtual space formed by code and not just the space that one is surrounded by. I have attempted to show that it is the shift to an architecture of code that has helped shape this changing world and which *Microserfs* tries to represent. Ultimately the different significations of code may not line up behind one another either. But as a form of response to the changes that they are living through, code can be about something other than work for the *Microserfs*: 'It's about all of us staying together' (199).

If this sentimental note seems to jar, then it should be remembered just what the 'all of us' contains in this novel. It is not just Dan and his girlfriend Karla, but it is gay Bug, bodybuilding coder Todd, Michael, mother and coder Dusty, Susan and Barcode too: a mixed gender and a mixed sexuality group of friends. And in a culture that has in the past so denigrated the closeness of the relationship between sons and their mothers, how revitalizing and how loving it is that, virtually paralysed after a stroke, Dan's mother is surrounded at her bedside by Dan and his friends. She can find only one way to communicate: 'part woman/part machine, emanating blue Macintosh light' she moves her fingers across a computer keyboard. Dan, anxious to confirm that it is his mother typing and not the machine, asks her a question that only she will be able to answer. 'Tell me something I never liked in my lunch bag at school'. She types 'PNUT BUTR'. 'Here it is', Dan says, 'Mom speaking like a license plate … like the lyrics to a Prince song … like encryption. All of my messing around with words last year and now, well … it's real life' (369). This sentimentality seems to me to be of a different order to that sentimentality that has so often been identified as a weakness; one subsequently used to denigrate and oppress women and homosexual men. It is instead one that, arising from a childish anxiety about transition, focuses on Dan and his friends' place within culture, how they cope with a culture that is often hostile to their nurture, and upon the reparative impulses that can help them to cope.

Notes

1. For a discussion of Microsoft and its internet products in relation to economic empire building see Lee and Fulford (2000).

2. Something of the flavour of this contemporary migration can be found in Po Bronson's account of Silicon Valley hopefuls who stake everything they have in their attempts to succeed in Silicon Valley. These 'Venture Trippers' arrive from Paris, from Salt Lake City, from Taiwan, from Boston, and share, according to Bronson, a sense of unbridled opportunity and excitement. Just as occurred in the 1840s, for most of the hopefuls the gamble does not pay off, of course, and they either filter into the many companies that

service the economics of Silicon Valley or have to return to the world from which they came. And yet there is always the one gambler who makes it big and who can sign a deal that will instantly make them a multi-millionaire and so ratify the economic culture of individual enterprise. Of the five hopefuls who Bronson follows for several months, only one makes it big. Ben Chiu sells his KillerApp program and company for $46.6 million, of which his share is fifty percent. Of the other four, two manage to keep their start-ups ticking over with small-scale venture capital, while the other two end up working for computer companies (Bronson, 1999: 3-39).

3. Emphasizing even further the importance of the office as an arena of play as well as work that is signified by the soccer ball, one of the other photographs in the series shows David Filo playing baseball in the open plan office with two of his colleagues.

4. For Bill Gates coding seems to have been something from which he has always made money. When he was twelve years old, he and Paul Allen—two years his senior—made $4, 2000 during the summer by writing a school scheduling programme (Cringely, 1996: 97).

5. In a vain attempt to find information and photographs about Microsoft's Redmond campus I tried to search the web using the keywords 'Microsoft' and 'architecture'. The results list produced only detailed articles about the structural syntax of Windows and other Microsoft software. The architecture of code, it seems, has become a whole business in itself.

6. I use this spelling of 'surveilling' intentionally, specifically drawing attention to its connection with the kinds of surveillance described by Michel Foucault. Foucault (1977/1991).

7. Although for the world's leading company it is still, in terms of staff, very small. It employed 17,800 people in 1995, the year *Microserfs* was published (Cusumano and Selby, 1996: 3). IBM, in contrast, employs something in the region of 380,000 (Cringeley, 1996: 121).

8. Each collection of eight digits represents a binary number that can be converted into a decimal number and from there into ASCII text. The original idea for this translation was taken from You (2000). I have amended the translated code published on this website to correct some errors.

References

Aranowitz, Stanley, Barbara Martinsons, and Michael Menser (eds.) (1996) *Technoscience and Cyberculture*, New York and London: Routledge.

Bronson, Po (1999) *The Nudist on the Late Shift: And Other Tales from Silicon Valley*, London: Secker & Warburg.

Ceruzzi, Paul E. (2000) *A History of Modern Computing*. Cambridge, Mass.: MIT Press.

Coupland, Douglas (1995) *Microserfs*, London: Flamingo.

Cringely, Robert X. (1996) *Accidental Empires*. Harmondsworth: Penguin.

Cusumano, Michael A., and Richard W. Selby (eds.) (1996) *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*, London: HarperCollins.

Foucault, Michel (1977/1991) *Discipline and Punish*, Harmondsworth: Penguin.

Gates, Bill (1996) *The Road Ahead*. Harmondsworth: Penguin. Revised Edition.

Guerlain, Pierre (2000) 'A Technofreak in Xanadu: Bill Gates and Cool Capitalism', *European Journal of American Culture*, 19(2): 116-22.

Kroker, Arthur (1996) 'Virtual Capitalism', in Stanley Aranowitz, Barbara Martinsons, and Michael Menser (eds.) *Technoscience and Cyberculture*, New York and London: Routledge.

Lee, Debbie and Fulford, Tim (2000) 'Virtual Empires', *Cultural Critique*, 44 (Winter): 3-28.

Leibovich, Mark (1996) 'Sleepless in Silicon Valley: Global Competition, Technology's Fast Track Dictate a Lifestyle of Working into the Wee Hours', *San Jose Mercury News* June 21, sec. Front: 1A.

Menser, Michael and Aronowitz, Stanley (1996) 'On Cultural Studies, Science and Technology', in Stanley Aranowitz, Barbara Martinsons, and Michael Menser (eds.) *Technoscience and Cyberculture*, New York and London: Routledge.

Moody, Fred (1995/1996*) I Sing the Body Electronic: A Year with Microsoft on the Multimedia Frontier*, London: Hodder and Stoughton.

Porter, Kevin J. (2000) 'Terror and Emancipation: The Disciplinarity and Mythology of Computers', *Cultural Critique*, 44 (Winter): 43-83.

Stoneley, Peter (1996) 'Rewriting the Gold Rush: Twain, Harte and Homosociality', *Journal of American Studies*, 30(2): 189-209.

You, Lawrence (2000) Microserfs Secret. 1st March 2000. Webpage. S. Chung. Available: http://coupland.tripod.com/msecret.html. 4th August.